

CLAIMS

What is claimed is:

1 1. A method comprising:

2 generating a function having an argument, the function expressed in a high-
3 level programming language, wherein the function includes a set of one or more
4 instructions that instruct a compiler unit to return a memory address of the argument
5 as a result of the function; and

6 generating a call to the function, the call expressed in the high-level
7 programming language, wherein the call causes the compiler unit to pass a descriptor
8 as the argument.

1 2. The method of claim 1, wherein the high-level programming language
2 includes a Fortran programming language.

1 3. The method of claim 1, wherein the argument is an integer data type.

1 4. A method comprising:

2 receiving a first code, wherein the first code refers to a variable of a target data
3 type, wherein the variable is addressable using a descriptor; and

4 translating the first code into a second code, the second code expressed in a
5 high-level programming language, wherein the translation requires a memory address
6 of the descriptor, and wherein the translation comprises:

7 generating a function having an argument, wherein the function
8 includes a set of one or more instructions that instruct a compiler unit to return the
9 memory address of the argument as a result of the function; and

10 generating a call to the function, wherein the call causes the compiler
11 unit to pass the descriptor as the argument.

1 5. The method of claim 4, wherein the translating comprises generating an
2 interface block for the function for each different target data type in the first code.

1 6. The method of claim 4, wherein the high-level programming language
2 includes a Fortran programming language.

1 7. The method of claim 4, wherein the argument is an integer data type.

1 8. The method of claim 4, further comprising generating a data structure to store
2 information based on the target data type.

1 9. The method of claim 7, wherein the function includes a routine from a runtime
2 library, the routine to return a memory address of an argument of the routine.

1 10. The method of claim 9, wherein the routine from the runtime library is written
2 in a C programming language.

1 11. A machine-readable medium that provides instructions, which when executed
2 by a machine, causes the machine to perform operations comprising:

3 generating a function having an argument, the function expressed in a high-
4 level programming language, wherein the function includes a set of one or more
5 instructions that instruct a compiler unit to return a memory address of the argument
6 as a result of the function; and

7 generating a call to the function, the call expressed in the high-level
8 programming language, wherein the call causes the compiler unit to pass a descriptor
9 as the argument.

1 12. The machine-readable medium of claim 11, wherein the high-level
2 programming language includes a Fortran programming language.

1 13. The machine-readable medium of claim 11, wherein the argument is an integer
2 data type.

1 14. A machine-readable medium that provides instructions, which when executed
2 by a machine, causes the machine to perform operations comprising:
3 receiving a first code, wherein the first code refers to a variable of a target data
4 type, wherein the variable is addressable using a descriptor; and
5 translating the first code into a second code, the second code expressed in a
6 high-level programming language, wherein the translation requires a memory address
7 of the descriptor, and wherein the translation comprises:
8 generating a function having an argument, wherein the function
9 includes a set of one or more instructions that instruct a compiler unit to return the
10 memory address of the argument as a result of the function; and
11 generating a call to the function, wherein the call causes the compiler
12 unit to pass the descriptor as the argument.
13

1 15. The machine-readable medium of claim 14, wherein the translating comprises
2 generating an interface block for the function for each different target data type in the
3 first code.

1 16. The machine-readable medium of claim 14, wherein the high-level
2 programming language includes a Fortran programming language.

1 17. The machine-readable medium of claim 14, wherein the argument is an integer
2 data type.

1 18. The machine-readable medium of claim 14, further comprising generating a
2 data structure to store information based on the target data type.

1 19. The machine-readable medium of claim 14, wherein the function includes a
2 routine from a runtime library, the routine to return a memory address of an argument
3 of the routine.

1 20. The machine-readable medium of claim 19, wherein the routine from the
2 runtime library is written in a C programming language.

1 21. A system comprising:

2 a translation unit to receive a first code that refers to a variable of a target data
3 type, wherein the variable is referred to by a descriptor, the translation unit to translate
4 the first code into a second code, the second code based on a high-level programming
5 language, wherein the translation requires a memory address of the descriptor and
6 wherein the translation comprises:

7 generating a function having an argument, wherein the function
8 includes a set of one or more instructions that instruct a compiler unit to return
9 the memory address of the argument as a result of the function; and

10 generating a call to the function, wherein the call causes the compiler
11 unit to pass the descriptor as the argument;

12 a compiler unit coupled with the translation unit, the compiler unit to receive
13 the second code and to generate object code, wherein the compiler unit passes the
14 descriptor as the argument based on the call to the function.

15 a linker unit coupled with the compiler unit to link the object code and a
16 number of routines to generate executable code, wherein the address of the descriptor
17 can be retrieved through one of the number of routines when the executable code is
18 executed.

1 22. The system of claim 21, wherein the generation of the second code includes
2 the generation of a function, the function having an entity as an argument, and a call
3 to the function, wherein the call to the function instructs a compiler unit to accept the
4 argument as an entity for which the memory address can be determined and returned
5 as a result of the function.

1 23. The system of claim 21, wherein the entity is an integer.

1 24. The system of claim 21, wherein the high-level programming language includes
2 a Fortran programming language.

- 1 25. The system of claim 21, wherein the function includes a routine from a
2 runtime library, the routine to return a memory address of an argument of the routine.
- 1 26. The system of claim 25, wherein the routine from the runtime library is written
2 in a C programming language.